



## Ciclo for

### Transcripción

[00:00] Ahora, yo he renombrado nuevamente el nombre de esta clase a EjemploCiclosWhile 2. ¿Por qué? Porque esta no es la única forma que tenemos de hacer una iteración, de hacer un ciclo. Tenemos otra forma más, que sería usando la palabra reservada for, que tendría una syntax diferente.

[00:24] Entonces vamos a crear otra clase, new class, EjemploCiclosFor, que sería esto. Le damos finish y vamos a crear nuestro método main. Vamos a seguir el mismo ejemplo anterior de CiclosWhile 1 por decirlo de alguna manera. Vamos a imprimir todos los números del 1 al 10 pero usando un operador diferente que sería el for.

[00:55] La sintaxis de for es un poco extraña, es una sintaxis que al ojo humano, a simple vista no es directamente entendible, es un poco complicada. No complicada, es un poco rara. Entonces, es simplemente for, de la misma forma tiene su paréntesis y tiene su scope, tiene su alcance.

[01:18] For aquí tiene tres áreas. Vamos a separar aquí, y normalmente sabemos que los puntos y comas van al final de cada statement, pero en este caso, en el de for, van aquí adentro. Entonces, la sintaxis de for es básicamente primero tenemos a la variable, declaramos una variable aquí que va a ser el tipo entero y va a ser contador, y contador es inicializado en 0.

[01:54] La segunda parte de aquí sería la condicional. La condicional sería: mientras que contador sea menor o igual a 10, vas a hacer esto. Y la tercera

parte del for es lo que vas a hacer al final de esta iteración. Por ejemplo vamos a escribir aquí arriba para entender cómo va esto para no olvidarlo.

[02:29] Esta es la variable, aquí está la condición y aquí es lo que se va a ejecutar al final de la iteración. Nosotros podemos tener aquí equis líneas de código y cuando esto se ejecute, haya terminado, ejecutamos esta condición de aquí. En este caso pues vamos a hacer que contador se incremente en 1. Perfecto. Entonces, aquí adentro vamos a decirle un sysout, `system.out.println` de contador. Listo.

[03:13] Entonces nuevamente, retomando, la primera parte de for es la declaración de la variable, la segunda es nuestra condición y la tercera es lo que vamos a ejecutar al final de que se haya ejecutado esta línea de código. Vamos a guardar aquí, vamos a ejecutar esto y tenemos todos los números nuevamente del 1 al 10.

[03:39] Hemos hecho lo mismo de aquí con while pero usando un ciclo diferente que sería for. Ahora, algo que puede no haber quedado muy claro puede ser esto de la variable, porque aquí yo estoy declarando una variable como 0. ¿Cómo funciona for? En la primera que él entra aquí a este ciclo él va a evaluar a contador igual a 0, contador menor o igual que 10 y al final se ejecuta esto y lo voy a incrementar.

[04:11] La siguiente vez, la segunda vez que él vuelva aquí a for él va a ignorar esto, él ya no va a volver a declarar contador igual a 0 porque si no, nunca se cumpliría esta condición y esto sería siempre true. Entonces él, a partir del segundo ciclo, él ignora esto y solamente evalúa la condición, él va a decir: "esto es true, es true". Entonces, ejecuto código, y al final ejecuto esto. Perfecto. Vuelvo aquí.

[04:44] Ignoro esto, y mientras esto sea true, ejecuto esta parte de aquí y al final ejecuto esta parte de aquí. Es así como se va a repetir todo eso. Ahora, si yo

desea solamente imprimir contador al final. ¿Será que yo puedo hacer todo esto? Contador. Vemos que no compila, porque contador no existe.

[05:14] ¿Por qué? Porque está dentro del scope de for. Fuera del contexto de for esta variable no existe y no es accesible desde afuera. ¿Cuándo tiene sentido entonces usar un for en lugar de un while? Simple. Cuando no necesitemos usar la variable que va a ser incrementada fuera del contexto de for.

[05:39] Por ejemplo, a diferencia de while, nosotros podemos usar al final el total porque nosotros queríamos en este caso hallar la suma de todos los números del 1 al 10, pero en el caso de for, nosotros no vamos a necesitar esta variable fuera, entonces simplemente aquí tendría más sentido usar esta sintaxis en lugar de un while clásico.